

## Addendum

1. TRANSLATOR FOR IP NETWORKS, NETWORK SYSTEM USING THE TRANSLATOR, AND IP NETWORK COUPLING METHOD THEREFOR

JC678 U.S. PTO

09/472838



12/28/99

TRANSLATOR FOR IP NETWORKS, NETWORK SYSTEM USING THE  
TRANSLATOR, AND IP NETWORK COUPLING METHOD THEREFOR

Inv. 91

BACKGROUND OF THE INVENTION

The present invention relates to an internet  
protocol (IP) network coupling method, a translator for IP  
5 networks, and a network system using the translator which  
are intended to couple a first class IP network in which a  
plurality of devices of interest are assigned IP addresses  
of a first class so as to prevent the assignment of the  
same IP address to different devices and a second class IP  
10 network in which a plurality of devices of interest are  
assigned IP addresses of a second class so as to prevent  
the assignment of the same IP address to different devices.

The protocol most widely known at present as a  
protocol for network layers used in the TCP/IP  
15 communication is the internet protocol (IP). The IP  
functions to provide an addressing service for specifying a  
destination to be communicated with among a large number of  
nodes connected to a network, and so on, which are the same  
services provided by the third layer of OSI (Open Systems  
20 Interconnection) reference model. Currently, the IP of  
version 4 (hereinafter called "IPv4") is commonly used.  
The IPv4 specification is disclosed in "INTERNET PROTOCOL;  
DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION", Information

Science Institute University of Southern California,  
September, 1981. A format defined for a header used in the  
IPv4 (hereinafter called "IPv4 header") is as shown in Fig.  
11B.

5           In the IPv4 header, a "version" field stores the  
version number, i.e., "4". A "header length" field stores  
the length of the IPv4 header itself. A "service type"  
field stores information indicative of a service quality of  
communication processing. A "packet length" field stores  
10 the size of an entire packet which includes a data block  
treated by the IP and the IPv4 header. Information  
directed down from an upper layer is treated as a single  
data block in the IP, and the data block is appended with  
an IPv4 header in the IP and then sent to a lower layer.  
15 Conversely, an IPv4 header included in a packet sent from a  
lower layer is analyzed by the IP, and a data portion of  
the packet is sent up to an upper layer depending on the  
analysis results. An "identifier" field stores an  
identifier used as reference information when data is  
20 passed to an upper layer. A "flag" field stores control  
information associated with fragmentation of a packet. A  
"fragment offset" field stores information indicating where  
fragmented data (fragment) has been positioned in original  
data. A "time to live" field stores a time during which an  
25 associated packet is allowed to exist in a network. A  
"protocol" field stores information indicating which  
protocol an upper layer employs. A "header checksum" field

stores a checksum for the IP header. A "source IP address" field stores the IP address of a source. A "destination IP address" field stores the IP address of a destination. The IP addresses are assigned to respective nodes connected to  
5 a network, and are set respectively to unique values in the network.

While the IP having the specification as mentioned above is currently spreading rapidly over a variety of communication services together with the  
10 increasing popularization of the internet, the IP is confronted with a grave problem of exhausted IP addresses.

As countermeasures for solving this problem, the IP of version 6 (hereinafter called "IPv6") has been proposed at present. The IPv6 specification is disclosed  
15 in S. Deering "INTERNET PROTOCOL: Version 6 (IPv6)", Xerox, PARC, December 1981.

A format defined for a header used in the IPv6 (hereinafter called "IPv6 header") is as shown in Fig. 11A. In the IPv6 header, a "version" field stores the version  
20 number, i.e., "6". A "priority" field stores the priority of processing executed by a router in a network for relaying a packet. A "flow label" field is used to store an identifier for performing a priority control or the like. A "payload length" field stores the length of a data  
25 portion calculated by subtracting an IPv6 header portion from a packet. A "next header identifier" field stores an identifier for identifying which header of upper layer

protocol, or which IPv6 extended header follows the IPv6 header. A "hop limit" field stores a maximum number of times of transfers for an associated packet. A "source IP address" field stores the IP address of a source, and a  
5 "destination IP address" field stores the IP address of a destination. Additionally, in the IPv6 header, the "source IP address" field and the "destination IP address" field are respectively extended from 32 bits to 128 bits. The extended address fields enable an increased number of nodes  
10 to be connected to a network.

#### SUMMARY OF THE INVENTION

Although the IPv6 has extended fields for storing the IP addresses, additional modifications partially added to the header and associated processing give rise to  
15 another problem that the IPv6 is not compatible with the existing IPv4.

From now on, the IPv6 will be implemented as the network protocol in nodes newly connected to a network, so that addresses conforming to the IPv6 will be assigned to  
20 these nodes as their network addresses. However, since simultaneous replacement of the previous network protocol with the IPv6 in all existing nodes is actually impossible, it is anticipated that coexistence of IPv4 nodes and IPv6 nodes will last for a long time. In addition, a portion of  
25 existing nodes (particularly peripherals), for which the replacement of the protocol is difficult, will continue to use the IPv4 as before.

5                   However, since the IPv6 and the IPv4 employ  
different header formats and so on from each other, IPv6  
nodes and IPv4 nodes cannot be simply coupled.

Referring specifically to Fig. 13, an IPv6 network 52 to which an IPv6 node (IPv6 host) 10 is connected and an IPv4 network 54 to which an IPv4 node (IPv4 host) 20 is connected are mutually connected through a translator 30. The IPv4 host 20 is designated an address uniquely assigned in the IPv4 network 54 (hereinafter called "IPv4 address"), while the IPv6 host 10 is designated both of an address uniquely assigned in the IPv6 network 52 (hereinafter called "IPv6 address") and an IPv4 address. The IPv6 host 10 represents itself with the IPv6 address when communicating with another IPv6 host (not shown) in the IPv6 network 52, and represents itself with an IPv4 address when communicating with the IPv4 host 20 in the IPv4 network 54. For example, for sending a packet from the IPv6 host 10 to the IPv4 host 20, the IPv6 host 10 stores its own IPv4 address (of the IPv6 host 10) in the "source IP address" field (see Fig. 11A) in a form as shown

in Fig. 12A, and stores the IPv4 address of the destination (the IPv4 host 20) in the "destination IP address" field (see Fig. 11A) in a form as shown in Fig. 12B, when generating a header for the packet to be sent. In Fig.

5 12A, the lower 32 bits within the 128 bits reserved for the "IP address" field are used for storing address information with the remaining bits being set at "0". The address represented in such a format is commonly called an "IPv4-compatible-IPv6 address". In Fig. 12B, in turn, the lower  
10 32 bits within the 128 bits reserved for the "IP address" field are used for storing address information, with 47th to 32nd bits being set at "1" and the remaining bits being set at "0". The address represented in such a format is commonly called an "IPv4-mapped-IPv6 address". Then, the  
15 header containing predetermined information and data to be sent is forwarded to the translator 30 as a single packet.

The translator 30 translates the packet sent thereto into a packet for the IPv4 network 54.

Specifically, the lower 32 bits, i.e., the IPv4 address of  
20 the IPv6 host 10 is retrieved from the foregoing IPv4-compatible-IPv6 address included in the header of the sent  
packet, and stored in the "source IP address field" of the  
IPv4 header shown in Fig. 11B. Simultaneously with this,  
the lower 32 bits, i.e., the IPv4 address of the source  
25 IPv4 host 20 is retrieved from the foregoing IPv4-mapped-IPv6 address included in the header of the sent packet, and stored in the "destination IP address" field of the IPv4

header shown in Fig. 11B. Subsequently, other necessary items are set in the header, and the header is sent together with the data to be sent to the IPv4 host 20 as a single packet.

5           The use of the foregoing method does enable an IPv6 node to interconnect with an IPv4 node. This method, however, needs to additionally assign an IPv4 address to an IPv6 node, thus causing a contradiction. The IPv6 addresses were introduced because an available number of  
10 IPv4 addresses had been lacking as mentioned above. Thus, if the IPv6 addresses intended to overcome this problem promoted the exhaustion of the IPv4 addresses by contraries, it would be meaningless to employ the IPv6  
*Summary* addresses.

15           In view of the problem as mentioned above, it is an object of the present invention to provide an IP network coupling method, a translator for IP networks, and a network system using the translator which are capable of coupling two networks employing different addressing  
20 architectures for IP addresses, due to a difference in IP version or the like, without exhausting IP addresses used in one of the two networks.

          According to one aspect of an IP network coupling method of the present invention to achieve the above  
25 object, there is provided a method of coupling IP networks for mutually coupling through a translator a first IP network, wherein a plurality of devices of interest are



assigned first IP addresses such that the same first IP address is not assigned to two or more devices of interest, and a second IP network, wherein a plurality of devices of interest are assigned second IP addresses such that the same second IP address is not assigned to two or more devices of interest, the method comprising the steps of:

when initiating a communication between a first device of interest which is one of a plurality of devices of interest existing in the first IP network and a second device of interest which is one of a plurality of devices of interest existing in the second IP network, assigning any of a plurality of previously prepared first IP addresses to a second IP address assigned to the second device of interest;

communicating between the second device of interest and the translator using the second IP address assigned to the second device of interest, and communicating between the translator and the first device of interest using the assigned first IP address; and releasing the assigned first IP address after termination of the communication.

According to one aspect of an IP network translator of the present invention to achieve the above object, there is provided a translator for coupling a first IP network, wherein a plurality of devices of interest are assigned first IP addresses such that the same first IP address is not assigned to two or more devices of interest, and a second IP network, wherein a plurality of devices of

a header translator which translates a header  
5 between a first IP packet used in the first IP network and  
a second IP packet used in the second IP network in order  
to exchange information between the first IP network and  
the second IP network; and

wherein, for a header translation performed to send information from the second IP network to the first IP network, any of the plurality of first IP addresses stored in the storage is assigned to a second IP address stored in a source storing field included in an IP header of the  
15 second IP packet, and the assigned first IP address is stored in a source storing field included in an IP header of the first IP packet; and

for a header translation performed to send  
20 information from the first IP network to the second IP  
network, the second IP address stored in the source storing  
field included in the IP header of the second IP packet is  
assigned to a first IP address stored in a destination  
storing field included in the IP header of the first IP  
25 packet, and the assigned second IP address is stored in a  
destination storing field included in the IP header of the  
second IP packet.

According to another aspect of the IP network translator of the present invention to achieve the above object, there is provided a network system comprising:

5 a translator for mutually coupling a first IP network, wherein a plurality of devices of interest are assigned first IP addresses such that the same first IP address is not assigned to two or more devices of interest, and a second IP network, wherein a plurality of devices of interest are assigned second IP addresses such that the  
10 same second IP address is not assigned to two or more devices of interest; and

a first device A of interest which is one of the plurality of devices of interest existing in the first IP network,

15 said first device A of interest comprising:

a storage which stores a plurality of first IP addresses different from each other;

an address translator which is operative when a first IP packet including data to be transmitted to a  
20 second device B of interest to the first network, the second device B of interest being one of a plurality of devices of interest existing in the second IP network, to assign any of the plurality of first IP addresses stored in the storage to a second IP address assigned to the second  
25 device B of interest, and store the assigned first IP address to a destination storing field included in an IP header of the first IP packet; and

a sender which sends address translation information including at least the second IP address assigned to the second device B of interest and the first IP address assigned to the second IP address, and

5           said translator comprising:

a storage which stores the address translation information sent from the first device A of interest; and

a packet translator which translates a packet between the first IP network and the second IP network  
10       using the address translation information.

Other objects, features and advantages of the present invention will become apparent from reading the following description of the embodiments of the invention taken in conjunction with the accompanying drawings.

15       BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic diagram illustrating an example of a network to which the present invention is applied;

Fig.2 is a block diagram illustrating functions  
20       of an IPv4/IPv6 translator for use in the network of Fig. 1;

Fig. 3A is an explanatory diagram illustrating the concept of processing performed in an IPv4 host of type A connected to the network of Fig. 1;

25       Fig. 3B is an explanatory diagram illustrating the concept of processing performed in an IPv4 host of type B connected to the network of Fig. 1;

Fig. 3C is an explanatory diagram illustrating the concept of processing performed in an IPv4 host of type C connected to the network of Fig. 1;

Fig. 4 is a block diagram illustrating functions of the IPv4 host of type A connected to the network of Fig. 1;

Fig. 5 is a flow chart representing a communication procedure (No. 1) applied to communications between an IPv6 host and an IPv4 host in the network of Fig. 1;

Fig. 6 is a flow chart representing a communication procedure (No. 2) applied to communications between an IPv6 host and an IPv4 host in the network of Fig. 1;

Fig. 7 is an explanatory diagram showing an example of an address translation information table provided in each of the IPv4 hosts of types A, B and the IPv4/IPV6 translator connected to the network of Fig. 1;

Fig. 8 is an explanatory diagram showing a format for a packet used to transmit information stored in the address translation information table of Fig. 7;

Fig. 9 is a schematic diagram illustrating another example of a network to which the present invention is applied;

Fig. 10A is a schematic diagram illustrating an exemplary hardware configuration of the IPv4/IPv6 translator connected to a network to which the present invention is applied;

Fig. 11A is an explanatory diagram of a format  
5 for an IPv6 header;

Fig. 11B is an explanatory diagram of a format for an IPv4 header;

Fig. 12A is an explanatory diagram of a format for an IPv4-compatible-IPv6 address;

10            Fig. 12B is an explanatory diagram of a format  
for an IPv4-mapped-IPv6 address; and

Fig. 13 is an explanatory diagram illustrating a conventional scheme for connecting an IPv4 network and an IPv6 network.

## 15 DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiments of the present invention will hereinafter be described with reference to the accompanying drawings.

As illustrated in Fig. 1, an embodiment of the present invention assumes an environment which allows coexistence of an IPv6 network 52 to which a plurality of IPv6 hosts 51 are connected and an IPv4 network 54 in which a plurality of IPv4 hosts 53 are connected, where these networks are interconnected through an IPv6/IPv4 translator 55 (hereinafter called "translator 55"). In the IPv6 network 52, a data is transferred through a packet having

the IPv6 header shown in Fig. 11A (hereinafter called "IPv6 packet"). In the header of the IPv6 packet, an IPv6 host 51 is represented by a normal IPv6 address, while an IPv4 host 53 is represented by an IPv4-mapped-IPv6 address (Fig. 12B). In the IPv4 network 54, a data is transferred through a packet having the IPv4 header shown in Fig. 11B (hereinafter called "IPv4 packet"). In the header of the IPv4 packet, the translator 55 is represented by an IPv4 address assigned by the translator 55 itself or by a particular IPv4 host 53, and each IPv4 host 53 is represented by a normal IPv4 address.

The translator 55, as illustrated in Fig. 2, comprises an IPv4/v6 reception processing unit 31 for sequentially fetching IPv6 packets flowing into the IPv6 network 52 and IPv4 packets flowing into the IPv4 network 54; a header translation unit 33 for translating the header of a packet fetched by the IPv4/v6 reception processing unit 31 based on address translation information stored in an address translation information table 35 and for updating the contents of the address translation information table 35 as required; an IPv4/v6 transmission processing unit 32 for sending a packet having a translated header to a network which is the destination of the packet; and an address translation information exchange unit 34 for exchanging the address translation information stored in the address translation information table 35 with address translation information stored in a particular node connected to the IPv4 network 54.

In this embodiment, there are three types (hereinafter called "type A", "type B", and "type C") of IPv4 hosts 53 connected to the IPv4 network 54. An IPv4 host 53 of type A is a host which allows for installation of application programs used in IPv6 hosts 51 (hereinafter called "IPv6 application programs") and has the aforementioned address translation information. A concept of processing performed by the IPv4 host 53 of type A is illustrated in Fig. 3A. An IPv4 host 53 of type C is a conventional host which has installed therein application programs used in IPv4 hosts 53 (hereinafter called "IPv4 application programs") as they are, and peripheral devices such as a printer fall under this type. A concept of processing performed by the IPv4 host 53 of type C is as illustrated in Fig. 3C. An IPv4 host 53 of type B is a host which has installed therein IPv4 applications and has the aforementioned address translation information. A concept of processing performed thereby is as illustrated in Fig. 3B. It should be noted that the particular node with which the translator 55 exchanges the address translation information is the IPv4 host 53 of type A or type B.

TCP/IPv4 processing 81 shown in Fig. 3A corresponds to general TCP/IP communication processing which provides services in conformity to the IPv4 in this embodiment. The TCP/IPv4 processing 81 employs a socket interface (IPv4 socket IF) as an interface with processing performed at a higher level. IPv6 application processing



84 is processing performed by an IPv6 application and employs a socket interface (IPv6 socket IF) as an interface with processing performed at a lower level, similarly to the TCP/IPv4 processing 81. Address translation processing 5 82 and address translation information exchange processing 83 respectively interpose between the IPv4 socket IF and the IPv6 socket IF. In the address translation processing 82, a data transfer including address translation processing is performed, while in the address translation 10 information exchange processing 83, the address translation information is exchanged between another node (for example, the translator 55) and the IPv4 host 53.

Fig. 4 illustrates the internal configuration of the IPv4 host 53 of type A.

15 A TCP/IPv4 reception processing unit 41 and a TCP/IPv4 transmission processing unit 42 are in charge of the TCP/IP processing 81. An address translation processing unit 43 is in charge of the address translation processing 82. An address translation information exchange 20 processing 44 is in charge of the address translation information exchange processing 83. In addition, the IPv4 host 53 of type A is provided with an address translation information table 45.

Next, description is made of a packet exchange 25 performed between the IPv6 host 51 and the IPv4 host 53 of type A.

First, the packet exchange originated by a communication from the IPv6 host 51 is described with

56 to the IPv6 network 52 (a1), the IPv6 host 51 sets its own IPv6 address "::1234:5678:9abc" as the source IP address, and the IPv4-mapped-IPv6 address of the IPv4 host 53 "::ffff:133.144.95.22" as the destination IP address in the header of the IPv6 packet 56.

The IPv4/v6 reception processing unit 31 of the translator 55 sequentially fetches IPv6 packets following through the IPv6 network 52, and determines, every time an IPv6 packet is fetched, whether or not the fetched IPv6 packet has an IPv4-mapped-IPv6 address stored in the "destination IP address" (specifically, a packet with the "destination IP address" field having 47th to 32nd bits set at "1" and all bits higher than them set at "0"). If the specified packet is found, this packet is sent to the header translation unit 33 (b1). The header translation unit 33, upon receiving the packet, extracts the IPv6 address, which is the source IP address, included in the packet, and searches out an IPv4 address which has previously corresponded to the extracted IPv6 address from the address translation information table 35 (b2). If the required IPv4 address does not exist in the address translation information table 35, the header translation unit 33 assigns a certain IPv4 address to the above-mentioned IPv6 address. In this case, an IPv4 address "192.168.10.3" is assigned to the IPv6 address "::1234:5678:9abc" (b3). The address translation information exchange unit 34 transmits the corresponding

000001: 00000000

relationship between these addresses to the IPv4 host 53 as address translation information (b4). A format for a packet used to exchange the address translation information is as shown in Fig. 8. In this embodiment,

5 "::<1234:5678:9abc" is stored in an "IPv6 address" field 101, and "192.168.10.3" is stored in an "assigned IPv4 address" field 102. An "option" field 103 shown in Fig. 8 can store a variety of control information required for communications, though not used specifically in this  
10 embodiment. It should be noted that the transmission of the address translation information at (b4) is performed not only to the IPv4 host 53 specified as a communication party but also all IPv4 hosts 53 having the address translation information table 45. Also, the header  
15 translation unit 33 stores the address translation information in the address translation information table (b5).

An exemplary structure of the address translation information table 35 is shown in Fig. 7. The shown address  
20 translation information table 35 is composed of an IPv6 address storing field 91; an assigned IPv4 address storing field 92; and optional storing field 93, which are filled in succession on a line-by-line basis. A plurality of IPv4 addresses to be assigned have previously been prepared and  
25 stored in a memory, not shown, in the translator 55. The address translation information table 35 itself is also stored in this memory.

Since a region in which assigned IPv4 addresses are used is closed in an associated IPv4 network, no problem will occur even if the same IPv4 address is used, for example, in a plurality of different IPv4 networks connected to a single IPv6 network 52. In other words, assuming that the IPv4 network 54 is an in-house communication network of a company and the IPv6 network 52 is an external communication network connected thereto through a public line, the company can assign IPv4 addresses which may possibly be used in an IPv4 network of any other company.

Subsequently, the header translation unit 33 replaces the source IP address in the packet from the IPv6 address "::1234:5678:9abc" with the IPv4 address "192.168.10.3". For the destination IP address, the IPv4 address extracted from the lower 32 bits of the IPv6 address is used. In addition, the header translation unit 33 simultaneously executes a variety of processing for translating the IPv6 header into the IPv4 header (b6). Subsequently, the IPv4/v6 transmission processing unit 32 sends the packet subjected to the translation processing at (b6) to the IPv4 host 53. If a corresponding address is found in the search processing at (b2), the found IPv4 address is employed as the source IP address, and therefore the processing at (b3), (b4), and (b5) are skipped.

The IPv4 host 53, upon receiving the address translation information sent from the translator 55 (c1), updates the contents of the address translation information

table 45 using the received address translation information (c2). This results in the contents of the address translation information table 35 in the translator 55 matching with the contents of the address translation information table 45 in the IPv4 host 53. The update of the contents of the address translation information table 45 is actually carried out by the address translation information exchange unit 44. The processing at (c1) and (c2) is performed not only for the IPv4 host which serves as a communication party but also for all IPv4 hosts.

Also, the IPv4 host 53, upon receiving the IPv4 packet sent from the translator 55 (c3), translates the address in the received IPv4 packet based on the updated address translation information table 45.

Specifically, the TCP/IPv4 reception processing unit 41 of the IPv4 host 53 receives the foregoing IPv4 packet flowing through the IPv4 network in accordance with TCP/IP, and passes the IPv4 packet to the address translation processing unit 43. The address translation processing unit 43, upon receiving the packet, extracts the IPv4 address "192.168.10.3", which is the source address of the packet, and searches out an IPv6 address which has corresponded to the extracted IPv4 address from the address translation information table 45. Since the contents of the address translation information table 45 have been updated at (c2), the IPv6 address "::1234:5678:9abc" is extracted here. The address translation processing unit 43

5  
10

15  
20  
25

5           The IPv4/v6 reception processing unit 31 of the  
translator 55 fetches an IPv4 packet 47 flowing through the  
IPv4 network 54 (b8), and passes the fetched IPv4 packet 57  
to the header translation unit 33. The header translation  
unit 33, upon receiving the packet, extracts the IPv4  
10 address "192.168.10.3", which is the destination IP address  
of the packet, and searches out an IPv6 address which has  
corresponded to the extracted IPv4 address from the address  
translation information table 35. Here, the IPv6 address  
"::1234:5678:9abc" is retrieved (b9). Subsequently, the  
15 header translation unit 33 sets an IPv4-mapped-IPv6 address  
"::ffff:133.144.95.22" of the IPv4 host 53 as the source IP  
address and the previously extracted IPv6 address  
"::1234:5678:9abc" as the destination IP address in the  
packet. In addition, the header translation unit 33  
20 simultaneously executes a variety of processing for  
translating the IPv4 header into the IPv6 header other than  
the address translation as mentioned above (b10). The  
IPv4/IPv6 transmission processing unit 32 sends the packet  
subjected to the translation processing at (b10) to the  
25 IPv6 host 51 (b11). Subsequently, the IPv6 host 51  
receives this packet (a2).

The foregoing IPv4 address which has corresponded to the IPv6 address may be released at the time a sequence

of communications have been terminated between the IPv6 host 51 and the IPv4 host 53. Also, an entry in the address translation information table may be deleted in accordance with a command or the like issued for system management of the network. Further, the option field of the address translation information table may be used to store a time period elapsed from the time an assigned IPv4 address is not used in communication such that the assigned IPv4 address may be forcedly released at the time a predetermined time has expired.

Next, a packet exchange will be described with reference to a flow chart of Fig. 6 for the case a communication is started from the IPv4 host 53.

Assume herein, similarly to the foregoing, that an IPv6 address "::1234:5678:9abc" has been assigned to the IPv6 host 51, and an IPv4 address "133.144.95.22" to the IPv4 host 53. The IPv6 application of the IPv4 host 53 sets its own IPv4 address "133.144.95.22" as the source IP address and sets the IPv6 address "::1234:5678:9abc" of the IPv6 host 51 as the destination IP address in a packet to be sent. This packet is passed to the address translation processing unit 43. The address translation processing unit 43, upon receiving the packet, extracts the IPv6 address, which is the destination IP address, included in the packet, and searches out an IPv4 address which has previously corresponded to the extracted IPv6 address from the address translation information table 45 (c1). If a



required IPv4 address does not exist in the address translation information table 45, the address translation processing unit 43 assigns a certain IPv4 address to the aforementioned IPv6 address. Assume herein that an IPv4 address "192.168.10.3" is assigned to the IPv6 address "::1234:5678:9abc" (c2). The address translation processing unit 43 transmits the corresponding relationship between these addresses to the translator 55 as address translation information (c3). A format for a packet used to exchange the address translation information is as shown in Fig. 8. The address translation processing unit 43 also stores the address translation information in the address translation information table 45 (c4). An exemplary structure of the address translation information table 45 is shown in Fig. 7. A plurality of IPv4 address to be assigned have previously been prepared and stored in a memory, not shown, in the IPv4 host 53. The address translation information table 45 itself is also stored in this memory. Subsequently, the address translation processing unit 43 replaces the IPv6 address "::1234:5678:9abc" in the packet with the IPv4 address "192.168.10.3". The source IP address is maintained as it is. In addition, the address translation processing unit 43 simultaneously executes a variety of processing for translating the IPv6 header into the IPv4 header other than the address translation as mentioned above (c5). Subsequently, the TCP/IPv4 transmission processing unit 42

sends the packet subjected to the translation processing in the address translation processing unit 43 (IPv4 packet 57 in Fig. 1) to the translator 55 (c6). If a corresponding IPv4 address is found in the search processing at (c1),  
5 this IPv4 address is employed as the source IP address, and therefore the respective processing at (c2), (c3), and (c4) are skipped.

The translator 55, upon receiving the address translation information sent from the IPv4 host 53 (c1),  
10 updates the contents of the address translation information table 35 using the received address translation information (b2). This results in the contents of the address translation information table 45 in the IPv4 host 53 matching with the contents of the address translation  
15 information table 35 in the translator 55. The update of the contents of the address translation information table 35 is actually carried out by the address translation information exchange unit 34. The transmission of the address translation information at (c3) is performed not  
20 only for the translator 55 but also for all IPv4 hosts 53 having the address translation information table 45, so that the contents of the address translation information tables 45 are updated in the respective IPv4 hosts 53. This prevents the same IPv4 address from being assigned to  
25 the respective IPv4 hosts 53 and to the translator 55.

Then, the IPv4/v6 reception processing unit 31 of the translator 55 fetches an IPv4 packet flowing through the IPv4 network 54 (b3), and passes the fetched IPv4

000001: 00000000

packet to the header translation unit 33. The header translation unit 33, upon receiving the packet, extracts the IPv4 address "192.168.10.3", which is the destination IP address of the packet, and searches out an IPv6 address  
5 corresponded to the extracted IPv4 address from the address translation information table 35. Since the contents of the address translation information table 35 have been updated at (b2), the IPv6 address "::1234:5678:9abc" is extracted here (b4). Subsequently, the header translation  
10 unit 33 sets an IPv4-mapped-IPv6 address "::ffff:133.144.95.22" of the IPv4 host 53 as the source IP address and the previously extracted IPv6 address "::1234:5678:9abc" as the destination IP address in the packet. In addition, the header translation unit 33  
15 simultaneously executes a variety of processing for translating the IPv4 header into the IPv6 header other than the address translation as mentioned above (b5). Then, the IPv4/IPv6 transmission processing unit 32 sends the packet subjected to the translation processing at (b5) to the IPv6  
20 host 51 (b6). Subsequently, the IPv6 host 51 receives this packet (a1).

Also, the IPv6 host 51 may send the IPv6 packet 51 to the IPv4 host 53 as application processing. In this event, the IPv6 host 51 sets its own IPv6 address  
25 "::1234:5678:9abc" as the source IP address and the "IPv4-mapped-IPv6 address "::ffff:133.144.95.22" of the IPv4 host 53 as the destination IP address in the header of the IPv6 packet 56, when sending the IPv6 packet 56 to the IPv6

network 52 (a2). This IPv6 packet 56 is sent to the IPv4 host 53 through the translator 55 (c7). Since the processing performed by the translator 55 (at (b7), (b8), (b9), (b10)) is similar to that of (b1), (b2), (b6), (b7) in Fig. 6, explanation thereof is not repeated here.

For the aforementioned IPv4 address assigned to the IPv6 address, the release processing may be performed in a manner similar to the aforementioned one.

Next, the IPv4 host 53 of type B will be described. As mentioned above, the IPv4 host 53 of type B is a host having an IPv4 application installed therein and possesses the aforementioned address translation information. The IPv4 host 53 of type B performs TCP/IPv4 processing 81, address translation processing 82, address translation information exchange processing 83, and IPv4 application processing 85, as illustrated in Fig. 3B. The IPv4 application processing 85 is executed on the TCP/IPv4 processing 81 through an IPv4 socket interface, as illustrated in Fig. 3B. The hierarchical structure of the processing is the same as that of the conventional IPv4 host 53, and in the IPv4 application processing 85, the IP address of a packet sent up from the TCP/IPv4 processing 81 is received by an IPv4 address. The TCP/IPv4 reception processing unit 41 and the TCP/IPv4 transmission processing unit 42 is in charge of the TCP/IPv4 processing 81. The address translation processing unit 43 is in charge of the address translation processing 82. The address translation

Thus, the application side does not particularly require an IPv6 address in the IPv4 host 53 of type B.

For user's convenience, the IPv4 host 53 of type B outputs the IPv6 address of a communication party on a display screen or the like. Specifically, the address translation processing unit 43 of the IPv4 host 53 of type B, upon receiving a packet outputted from the IPv4 application, extracts an IPv4 address, which is the source IP address of the packet, searches out an IPv6 address corresponded to the extracted IPv4 address from the address translation information table 45, and outputs the IPv6 address on the display screen or the like.

Further, the IPv4 host 53 of type B allows the user to arbitrarily make correspondence between the IPv6 address and the IPv4 address which is automatically performed in the translator and the IPv4 host of type A. Specifically, the address translation processing unit 43 updates the address translation information table 45 in accordance with the contents of operation performed by the user. For example, if the user assigns an IPv4 address "192.168.10.3" to an IPv6 address ":::1234:4567:9abc", the address translation information table 45 is set to the contents, for example, as shown in Fig. 7. After the address translation information table 45 has been updated,

the address translation information exchange unit 34 creates a packet including the updated portion (see Fig. 8) and transmits the packet to other IPv4 hosts 53 which possess the address translation information tables. This results in the contents of the address translation information tables 45 in the respective IPv4 hosts 53 matching with the contents of the address translation information table 35 in the translator 55.

Since a packet exchange performed between the IPv6 host 51 and the IPv4 host 53 of type B is achieved by a processing flow similar to that described previously with reference to Figs. 5 and 6, explanation thereof is omitted.

Next, the IPv4 host 53 of type C will be described. The IPv4 host 53 of type C is, as mentioned previously, a host which has installed therein an IPv4 application used in the IPv4 host 53 as it is. A host which has an IPv4 application implemented in the form of ROM and therefore cannot modify the IPv4 application by software without external aid is also included in the type C host 53. In the IPv4 host 53 of type C, IPv4 application processing 85 is executed on TCP/IPv4 processing 81 through an IPv4 socket interface, as illustrated in Fig. 3C. The TCP/IPv4 reception processing unit 41 and the TCP/IPv4 transmission processing unit 42 is in charge of the TCP/IPv4 processing 81.

It should be noted that although the IPv4 host 53 of type C is not allowed to initiate a communication due to its configuration, communications can be initiated from the

other party to the IPv4 host 53 of type C without problems as described below.

The IPv4 application in the IPv4 host 51 of type C, upon receiving a packet outputted from the TCP/IPv4 reception processing unit 41, extracts the source IP address and the destination IP address of the received packet, and performs predetermined processing specified by a data portion of the received packet. Subsequently, the IPv4 application packs the processing results in the packet, if required, and passes the packet to the TCP/IPv4 transmission processing unit 42. In this event, the IPv4 application replaces the extracted destination IP address and source IP address with each other and sets them in the packet. For example, when an IPv4 address "133.144.95.22" is stored in the "destination IP address" field of the received packet and an IPv4 address "192.168.10.3" is stored in the "source IP address" field of the same, the IPv4 address "192.168.10.3" is stored in the "destination IP address" field and the IPv4 address "133.144.95.22" is stored in the "source IP address" field. This packet is sent to the IPv4 network 54 by the TCP/IPv4 transmission processing unit 42 to be delivered to the translator 55.

Thus, the embodiment has been described for the case where the IPv4 network and the IPv6 network are connected through the IPv4/v6 translator, where actual hardware of the IPv6/v4 translator used herein is configured, for example, as illustrated in Fig. 10A.

The IPv6/v4 translator is composed of a CPU71, a memory 72, and network interfaces 73-1, 73-2, ..., 73-n. The CPU 71 serves to manage the memory 72 and control the network interfaces 73-1, 73-2, ..., 73-n. The memory 72 previously stores a variety of programs which are executed by the CPU 71 as required to implement the IPv4/v6 reception processing unit 31, the IPv4/v6 transmission processing unit 32, the header translation unit 33, the address translation information exchange unit 34, and the address translation information table 35, all illustrated in Fig. 2.

It should be noted that since one each of the IPv6 network 51 and the IPV4 network 54 exist in Fig. 1, the network interface 73-1 is used for IPv6 network 52, and the network interface 73-2 is used for the IPv4 network 54, by way of example. In this case, the network interface 73-1 fetches an IPv6 packet flowing through the IPv6 network 52 and stores it in the memory 72, while the network interface 73-2 sends an IPv4 packet, produced by a header translation and so on, to the IPv4 network 54. When a packet is sent in the opposite direction, the network interfaces 73-1, 73-2 perform operations reverse to the foregoing. If a plurality of IPv6 and IPv4 networks exist, a number of network interfaces will be used corresponding to the number of networks.

Actual hardware of the IPv4 host is configured, for example, as illustrated in Fig. 10B. The IPv4 host is



composed of a CPU 71, a memory 72, and a network interface 73. The CPU 71 serves to manage the memory 72 and control the network interface 73. The memory 72 previously stores a variety of programs. For example, in the IPv4 host of  
5 type A, the programs in the memory 72 are executed by the CPU 71 as required to realize the TCP/IPv4 reception processing unit 41, the TCP/IPv4 transmission processing unit 42, the address translation processing unit 43, the address translation information exchange processing unit  
10 44, the address translation information table 45, and the IPv4 application. The network interface 73 fetches an IPv4 packet flowing through the IPv4 network 54 and stores it in the memory 72 as well as sends an IPv4 packet, produced by an address translation and so on, to the IPv4 network 54.

15 For a particular IPv6 host, an IPv4 address may have been previously assigned thereto and stored in the address translation information table. In this way, a processing time required to the assignment is reduced.

It will be understood that the present invention  
20 is not limited to the network configuration as illustrated in Fig. 1. For example, the present invention may also be applied to a network system which permits coexistence of an IPv4 network and an IPv6 network. Fig. 9 illustrates an IPv4/IPv6 mixed network 64 to which an IPv4 host 62 and an  
25 IPv6 host 61 are connected. While the IPv4/v6 mixed network 64 has coexistent IPv4 packets 66 and IPv6 packets 65, an IPv4/v6 translator 63 fetches these packets thereinto, performs the foregoing address translation and

header translation for the fetched packets, if required,  
and returns the processed packets to the network. This  
enables the IPv4 host 62 and the translator 63 to  
communicate through the IPv4 packet, and the translator 63  
5 and the IPv6 host 61, to communicate through the IPv6 packet  
65.

Further, the present invention is not limited to  
the coupling of an IPv4 network and an IPv6 network, but  
may be applied to two kinds of networks (a first class IP  
10 network and a second class IP network) which are different  
in IP addressing architecture due to a difference in  
version or the like.

The software programs performing the processings  
for the translation as shown in Figs. 5 and 6 may be stored  
15 in a computer-readable recording medium such as a  
semiconductor memory or a recording disk.

According to the present invention, at the time a  
communication is initiated between an device of interest A  
which is one of a plurality of devices of interest  
20 belonging to a first class IP network (for example, the  
IPv4 network) and an device of interest B which is one of a  
plurality of devices of interest belonging to a second  
class IP network (for example, the IPv6 network), a  
translator disposed between the first class IP network and  
25 the second class IP network or the device of interest A  
assigns an IP address of a first class to an IP address of  
a second class assigned to the device of interest B, such  
that the communication is carried out using this IP address

It is therefore possible to eliminate useless addressing, i.e., the previous assignment of an IP address of the first class to the device of interest B belonging to the second class IP network for communicating with the device of interest A belonging to the first class IP network, thus effectively utilizing a small number of IP addresses of the first class.

Also, the IP address of the first class assigned to the device of interest A is used only within the first class network, so that even if the same IP address is used in other external networks, any problem will not occur due to the duplicated IP address, thereby making it possible to more effectively utilize the IP addresses of the first class.